

AI ASSISTED CODING LAB

ASSIGNMENT-10.2

Name: Guangsinlung Phaomei

Enroll no: 2503A51L20

Batch: 19

TASK DESCRIPTION 1:

- Write python program as shown below.
- Use an AI assistant to review and suggest corrections.

```
def calcFact(n):
    result=1
    x=0
    for i in range(1,n):
        result=result*i
    return result

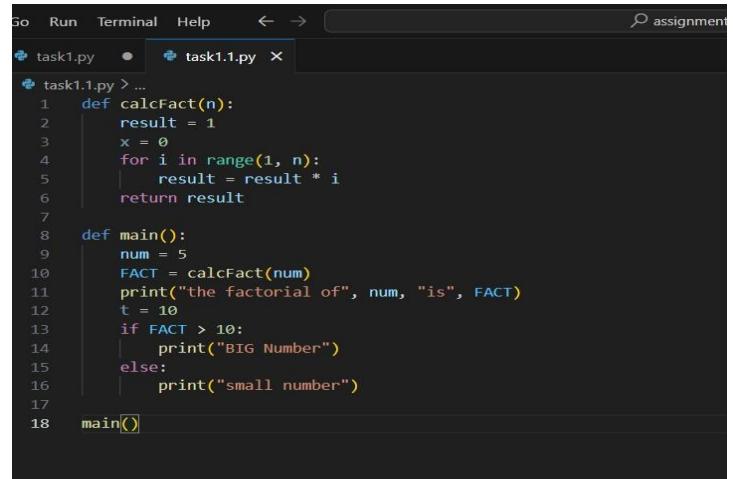
def main():
    num = 5
    FACT = calcFact(num)
    print("the factorial of",num,"is",FACT)
    t=10
    if FACT>t:
        print("BIG Number")
    else:
        print("small number")

main()
```

PROMPT 1:

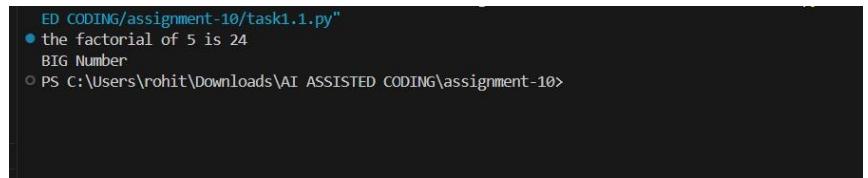
Generate a Python program that calculates the factorial of a number using a function. Then, use a conditional statement to print whether the result is a "BIG Number" or a "small number" based on a threshold.

CODE GENERATED:



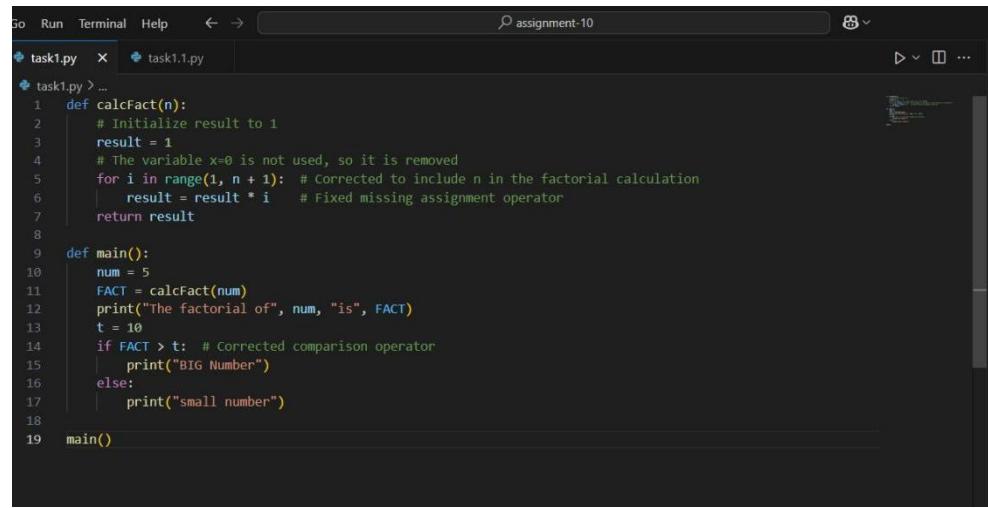
```
Go Run Terminal Help ← → ⌂ assignment
task1.py task1.1.py
task1.1.py > ...
1 def calcFact(n):
2     result = 1
3     x = 0
4     for i in range(1, n):
5         result = result * i
6     return result
7
8 def main():
9     num = 5
10    FACT = calcFact(num)
11    print("the factorial of", num, "is", FACT)
12    t = 10
13    if FACT > 10:
14        print("BIG Number")
15    else:
16        print("small number")
17
18 main()
```

OUTPUT:



```
ED CODING/assignment-10/task1.1.py"
● the factorial of 5 is 24
BIG Number
○ PS C:\Users\rohit\Downloads\AI ASSISTED CODING\assignment-10>
```

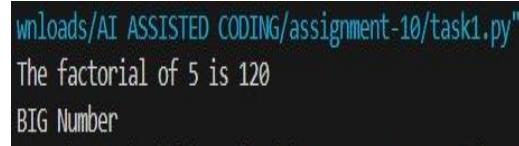
CORRECT CODE GENERATED:



The screenshot shows a code editor window titled "assignment-10". It has two tabs open: "task1.py" and "task1.1.py". The "task1.py" tab is active and contains the following Python code:

```
1  def calcFact(n):
2      # Initialize result to 1
3      result = 1
4      # The variable x=0 is not used, so it is removed
5      for i in range(1, n + 1): # Corrected to include n in the factorial calculation
6          result = result * i    # Fixed missing assignment operator
7      return result
8
9  def main():
10     num = 5
11     FACT = calcFact(num)
12     print("The factorial of", num, "is", FACT)
13     t = 10
14     if FACT > t: # Corrected comparison operator
15         print("BIG Number")
16     else:
17         print("small number")
18
19 main()
```

OUTPUT:



```
wnloads/AI ASSISTED CODING/assignment-10/task1.py"
The factorial of 5 is 120
BIG Number
```

OBSERVATION:

I created a Python program using functions, loops, and conditionals to calculate factorials. With AI help, I improved logic and readability, showing a clear understanding of Python basics.

TASK DESCRIPTION 2:

Write the Python code for Fibonacci as shown below and execute.

Ask AI to improve variable names, add comments, and apply PEP8 formatting (cleaned up).

Students evaluate which suggestions improve readability most. one.

```

def f1(xx):
    a=0
    b=1
    c=2
    Zz=[a,b]
    while c<=xx:
        d=a+b
        Zz.append(d)
        a=b
        b=d
        c=c+1
    return Zz

def m():
    NN=10
    ans=f1(NN)
    print("fib series till",NN,":",ans)

m()

```

PROMPT 1:

Write a Python program using a function to generate the Fibonacci series up to a given number of terms, then review the code with an AI assistant.

```

Go Run Terminal Help ← → assignment-10
task1.py task2.py task2.1.py task1.1.py
task2.1.py > f1
1 def f1(xx):
2     a = 0
3     b = 1
4     c = 2
5     Zz = [a, b]
6     while c <=xx:
7         d = a + b
8         Zz.append(d)
9         a = b
10        b = d
11        c = c + 1
12    return Zz
13
14 def m():
15     NN = 10
16     ans = f1(NN)
17     print("fib series till", NN, ":", ans)
18
19 m()

```

CODE GENERATED:

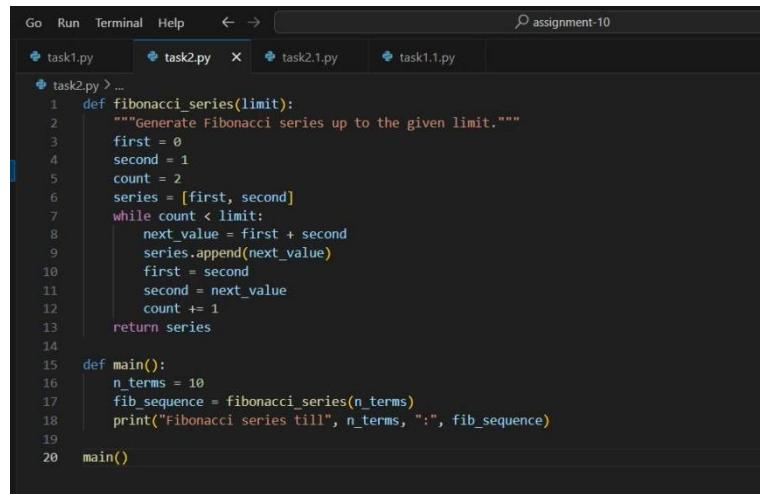
OUTPUT:

```

$ python task2.1.py
fib series till 10 : [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]

```

COREECT CODE GENERATED:



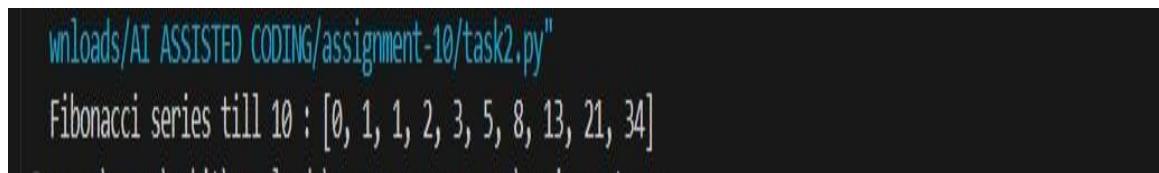
The screenshot shows a code editor window titled "assignment-10". The tabs at the top are "task1.py", "task2.py" (which is the active tab), "task2.1.py", and "task1.1.py". The code in "task2.py" is as follows:

```
def fibonacci_series(limit):
    """Generate Fibonacci series up to the given limit."""
    first = 0
    second = 1
    count = 2
    series = [first, second]
    while count < limit:
        next_value = first + second
        series.append(next_value)
        first = second
        second = next_value
        count += 1
    return series

def main():
    n_terms = 10
    fib_sequence = fibonacci_series(n_terms)
    print("Fibonacci series till", n_terms, ":", fib_sequence)

main()
```

OUTPUT:



The terminal window shows the command "python3 task2.py" being run. The output is:

```
wlloads/AI ASSISTED CODING/assignment-10/task2.py"
Fibonacci series till 10 : [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

OBSERVATION:

I created a Python program to generate the Fibonacci series using a function and loop. With AI feedback, I improved variable naming, formatting, and code clarity, enhancing readability and showing strong understanding and improvement through feedback

TASK DESCRIPTION 3:

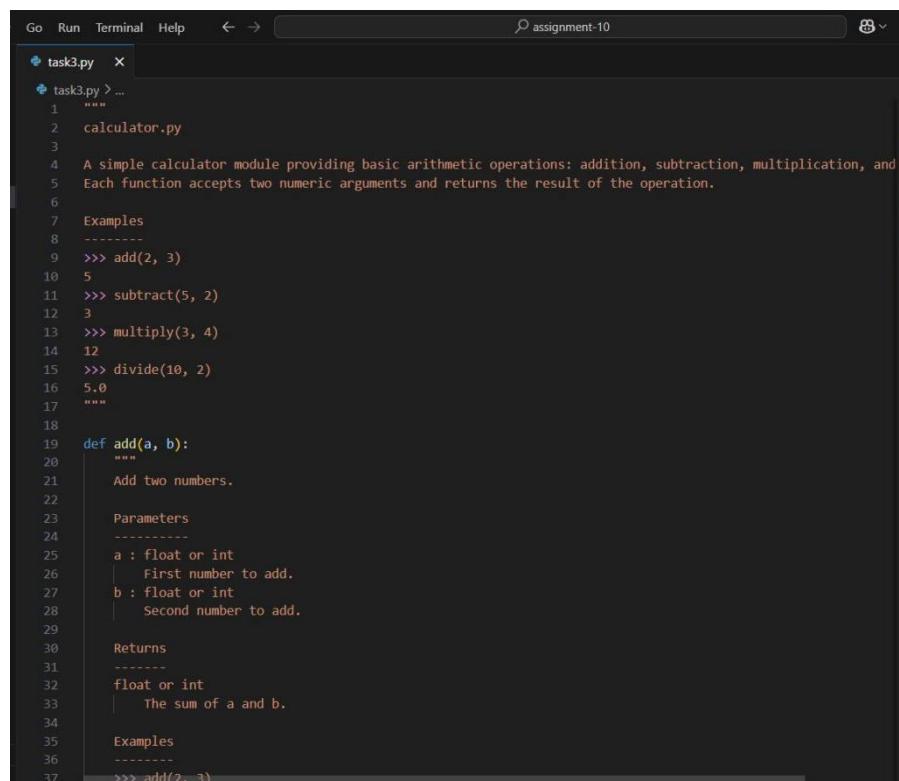
- Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide).
- Incorporate manual **docstring** in code with NumPy Style
- Use AI assistance to generate a module-level docstring + individual function docstrings.

- Compare the AI-generated docstring with your manually written one.

PROMPT 1:

Generate a Python program that does basic math (like adding, subtracting, multiplying, dividing) and practice writing clear documentation for your code.

CODE GENERATED:



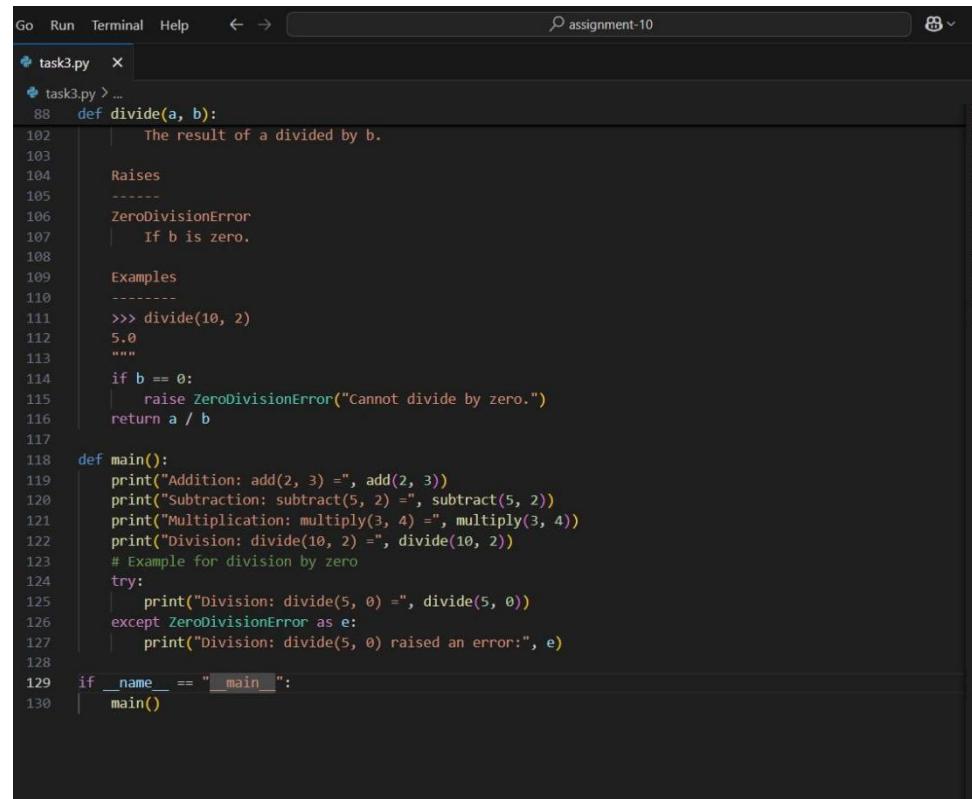
```

Go Run Terminal Help ← → | assignment-10
task3.py X
task3.py > ...
1 """
2     calculator.py
3
4     A simple calculator module providing basic arithmetic operations: addition, subtraction, multiplication, and
5     division. Each function accepts two numeric arguments and returns the result of the operation.
6
7 Examples
8 -----
9 >>> add(2, 3)
10 5
11 >>> subtract(5, 2)
12 3
13 >>> multiply(3, 4)
14 12
15 >>> divide(10, 2)
16 5.0
17 """
18
19 def add(a, b):
20     """
21         Add two numbers.
22
23     Parameters
24     -----
25     a : float or int
26         First number to add.
27     b : float or int
28         Second number to add.
29
30     Returns
31     -----
32     float or int
33         The sum of a and b.
34
35 Examples
36 -----
37 >>> add(2, 3)

```

```
Go Run Terminal Help ← → 🔍 assignment-10
task3.py ✘
task3.py > ...
19 def add(a, b):
20     """
21     >>> add(2, 3)
22     5
23     """
24     return a + b
25
26 def subtract(a, b):
27     """
28     Subtract one number from another.
29
30     Parameters
31     -----
32     a : float or int
33         Number to subtract from.
34     b : float or int
35         Number to subtract.
36
37     Returns
38     -----
39     float or int
40         The result of a minus b.
41
42     Examples
43     -----
44     >>> subtract(5, 2)
45     3
46     """
47     return a - b
48
49 def multiply(a, b):
50     """
51     Multiply two numbers.
52
53     Parameters
54     -----
55     a : float or int
56         First number to multiply.
```

```
Go Run Terminal Help ← → 🔍 assignment-10
task3.py ✘
task3.py > ...
65 def multiply(a, b):
66     """
67     a : float or int
68         Second number to multiply.
69
70     Returns
71     -----
72     float or int
73         The product of a and b.
74
75     Examples
76     -----
77     >>> multiply(3, 4)
78     12
79     """
80     return a * b
81
82 def divide(a, b):
83     """
84     Divide one number by another.
85
86     Parameters
87     -----
88     a : float or int
89         Numerator.
90     b : float or int
91         Denominator.
92
93     Returns
94     -----
95     float
96         The result of a divided by b.
97
98     Raises
99     -----
100    ZeroDivisionError
101    If b is zero.
```



```
Go Run Terminal Help ← → assignment-10
task3.py x
task3.py > ...
88 def divide(a, b):
102     """The result of a divided by b.
103     Raises
104     -----
105     ZeroDivisionError
106     | If b is zero.
107
108     Examples
109     -----
111     >>> divide(10, 2)
112     5.0
113     """
114     if b == 0:
115         raise ZeroDivisionError("Cannot divide by zero.")
116     return a / b
117
118 def main():
119     print("Addition: add(2, 3) =", add(2, 3))
120     print("Subtraction: subtract(5, 2) =", subtract(5, 2))
121     print("Multiplication: multiply(3, 4) =", multiply(3, 4))
122     print("Division: divide(10, 2) =", divide(10, 2))
123     # Example for division by zero
124     try:
125         print("Division: divide(5, 0) =", divide(5, 0))
126     except ZeroDivisionError as e:
127         print("Division: divide(5, 0) raised an error:", e)
128
129 if __name__ == "__main__":
130     main()
```

OUTPUT:

```
wloads/AI ASSISTED CODING/assignment-10/task3.py"
Addition: add(2, 3) = 5
Subtraction: subtract(5, 2) = 3
Multiplication: multiply(3, 4) = 12
Division: divide(10, 2) = 5.0
Division: divide(5, 0) raised an error: Cannot divide by zero.
```

OBSERVATION:

This assignment combines Python function design with good documentation practices. It reinforces core concepts like functions and parameters while promoting clear, professional NumPy-style docstrings. Comparing manual and AI-generated docstrings improved understanding of clarity and formatting.